

Compile Plugins and build Install Packages

This page holds instructions on how to compile plugins in a “standalone” environment and on how to create the plugin packages in Windows. The plugin watchdog_pi is used as an example, since this plug_in has no additional dependencies.

Prerequisites

Set up a working development environment as described on [Compiling on Windows](#), including all [prerequisites](#). Make sure that you're able to create an install package for OpenCPN successfully. After that, compiling plugins and building install packages for the plugins is easy.

Getting and building the plugin sources

- Open a **x86 Native Tools Command Prompt for VS 2017** (In Windows 10 by typing in the search box next to the Windows start button **x86 Native...**etc) and execute the following commands:

```
cd C:\Users\%username%\Sources
git clone https://github.com/seandepagnier/watchdog_pi
```

- Read README in your git clone source, which should list any other required dependencies. Download and install any required dependency into C:\Users\%username%\Sources\packagename_pi\source\buildwin.

```
cd C:\Users\%username%\Sources\watchdog_pi
more README
```

- In this particular case, the watchdog plugin does not have any other dependencies.
- Copy the file opencpn.lib that is created while building OpenCPN into the build folder of the plugin:

```
cd C:\Users\%username%\Sources\watchdog_pi
mkdir build
cd build
copy c:\Users\%username%\Sources\OpenCPN\build\release\opencpn.lib .
```

One should use the earliest version of **opencpn.Lib** possible that is still compatible with the Plugin API, according to the chart for [Plugin API and OpenCPN Version Compatibility](#).

- Build the plugin and create the install package via the following commands, waiting for successful completion of each command.

```
cd c:\Users\%username%\Sources\watchdog_pi\build
```

```
cmake -T v141_xp ..
cmake --build .
cmake --build . --config release
cmake --build . --config release --target package
```

If successful, CMake will return this message:

```
CPack: - package:
**C:/Users/%username%/Sources/watchdog_pi/build/watchdog_pi-2.4.0-ov50-
win32.exe** generated.
```

Execute this file to install the plugin via the mouse in the explorer, or by starting via the command line

```
C:/Users/%username%/Sources/watchdog_pi/build/watchdog_pi-2.4.0-ov50-win32.exe
```

Go through the installations steps, start OpenCPN and see if the plugin is available. Otherwise have a look in the logfile.

Configure NSIS



Can the information below be removed or moved to a page for older versions?
Currently there is NSIS 3.04 - KarmaSailor. 20191110

With NSIS 2.4.6 some registry tweaking was required. Go to “NSIS (Unicode) used to build the installation packages for OpenCPN and Plugins

1. Get NSIS from [Downloads - unsis - NSIS branch that supports Unicode. - Google Project Hosting](#)
2. The NSIS Unicode 2.46 setup file is also included in the GIT repository →
..\buildwin\NSIS_Unicode\nsis-2.46-Unicode-setup.exe
3. Direct download link:
<https://code.google.com/p/unsis/downloads/detail?name=nsis-2.46.3-Unicode-setup.exe&can=2&q=>
4. Download and install it.
5. Due to a “bug” in CMake, which only looks at “**HKEY_LOCAL_MACHINE\SOFTWARE\NSIS**” for the installation location of NSIS and the Unicode version adds its registry key in “**HKEY_LOCAL_MACHINE\SOFTWARE\NSIS\Unicode**”, there is some registry tweaking needed.
6. Previously I had not installed this, so I had to do it and I also ran the Batch file mentioned in the Dev Docs that they say to run to make the registry tweaks. **So run that batch file.**
7. Set your **Environment Path Variable** under **Control Panel > System > Advanced System Settings > Environment Variables > System Variable > Variable = Path** under **Values** add the path to your NSIS programs EG: “C:\Program Files (x86)\NSIS\Unicode with a colon in front. Be careful with your editing here.
8. **Then reboot.**

From:
<https://opencpn.org/wiki/dokuwiki/> - **OpenCPN Manuals**

Permanent link:
https://opencpn.org/wiki/dokuwiki/doku.php?id=opencpn:developer_manual:developer_guide:compiling_windows:compiling_external_plugins_and_building_install_packages

Last update: **2019/11/11 21:08**

